

REMARKS

7/23/03

Applicants have carefully examined the Office Action and this response is believed to address all issues raised in the Office Action.

Request for a Telephone Interview

Applicant respectfully requests the Examiner to grant a telephone interview prior to mailing a second office action so that outstanding issues could be discussed.

Request to Assign this Case to a Customer Number

Please assign this case to Customer Number 32127 and direct all future correspondence in this case to the correspondence address associated with Customer Number 32127.

Amendments to the claims

Claims 1, 7 and 19 are amended to overcome objections raised in the Office Action.

Claim 1 is amended to clarify that the term "application program" should read -- application service program--.

Claim 7 is amended to rectify a typographical error by inserting the proposition -- on--between "based" and "a criterion".

Claim 19¹ is amended to clarify that the term "the middleware component" should read --the object-oriented middleware component--.

The Office Action pointed out the need for these changes. No new matter is added as a result of these changes. Examiner is respectfully requested to review and enter these changes.

Claim Rejections Under 35 U.S.C. §103(a)

The Office Action rejected all pending claims under 35 U.S.C. §103(a) as being obvious over combinations of references. The Office Action rejected claims 1, 2, 4, 6-12, 14, 16 and 17 over Pospíšil et al in view of Hightower (USP 6,510,550 B1).

The Office Action stated that all the elements of claim 1 were present in Pospíšil except for memory for queuing customer/client request. The Office Action further stated that Hightower taught a memory for queuing customer/client request. Based on these

¹ The Office Action stated that this was claim 18, but Applicants believe that it is claim 19 that had the problem with antecedent basis.

statements, the Office Action reasoned that it would have been obvious to one skilled in the art to apply Hightower to the teachings of Pospisil. Applicants respectfully disagree with these statements and the reasoning based on these statements.

Reservation of Rights to Antedate Reference

First, because the office action claims that Pospisil was published in 1999, Applicants reserve the right to file an antedating affidavit swearing behind Pospisil should such a need arise.

The Claimed Invention

Second, Pospisil does not teach the elements of independent claim 1. In particular, Pospisil does not teach the following elements of claim 1.

a service manager bean coupled to said application service program for creating and returning to said client program a handle to a functional bean appropriate to the client request;

a data store interface for coupling said application service program to a data storage system; and

memory coupled to said application service program, said memory for queuing customer requests and to service the queued customer requests in accordance with the code contained in the functional bean and for interfacing with the data storage system via the data store interface.

Nothing in Pospisil discloses a functional bean. A functional bean is described in the instant application as follows.

"[I]t has been discovered that mapping an object to a function, rather than to a data element provides significant advantages, especially in managing limited resources that are accessed by a plurality of clients. Further, the presently available Enterprise JavaBeans (EJB) can be programmed to achieve such a mapping.

"Accordingly, in an embodiment, the disclosure is directed to a novel 'functional' bean, which is devoted to modeling a business function. Clients do not need to know the particular primary key or identifier as in the case of an entity EJB; rather a client knows only a well-known

Service Manager bean to obtain a handle to the correct type of functional bean.

"If a client needs to request a service offered by such a functional bean, it can be invoked directly. In one aspect, these functional beans are created and managed by a bean container such as an EJB container. The container controls access to these beans by way of a Service Manager. In one embodiment, the Service Manager itself is modeled as a functional bean, whose function is that of a manager of the other functional beans." See p. 7, ll. 18-30.

As to the differences between the then-existing state of the art and the instantly claimed invention, the Applicants stated as follows.

"This is based on the idea that the entity bean must model only entities (i.e., one or more database records). A problem with this approach is that the service providers are not visible to the client, where there could be a need for such visibility. As stated earlier, if there is a need for a number of database records to be simultaneously accessed by a client, then there will be a proliferation of such beans if each record is modeled as an entity bean. In extreme cases, such a proliferation may result in a depletion of available resources for other tasks. Further, since access to these entity beans should be serialized, each successive client with a need to access a particular row (or an entity bean) for an update should wait for its turn. Moreover, there is an implicit assumption that the client should know beforehand the primary key (or the identifier) of the entity bean it needs to contact. The current methods do not provide an easy solution to these problems, though one could craft an unwieldy solution using session beans. Accordingly, there is a need for a new bean type that provides transactional capability of an entity bean without its persistence, and models business functions." See p. 7, ll. 1-14.

Functional bean is further elaborated on pages 10-11. Thus, claim 1, which describes a computer system using a functional bean, includes this description of the functional bean. Further details of implementation of functional bean are also described in the section Detailed Description. Pospisil does not describe, for example, using an entity bean to model business functions, instead of database records, and which "provides transactional capability of an entity bean without its persistence."

Pospíšil describes the Enterprise JavaBean types as follows.

3.2 Session beans

As stated in the EJB specification, an EJB server has to support the session bean abstraction (unlike the entity beans that are optional). A session bean represents a "function", being by default temporary (*short-lived*), non-persistent, and thus of limited transactional capabilities (compared to the entity beans). The focus should be put especially on the following points:

- The "one transaction at a time" restriction.
- The implementation of the SessionContext interface.
- The functionality of the SessionSynchronization interface.
- The correct lifecycle implementation of both the stateful and the stateless session beans.

3.3 Entity beans

By default, an entity bean represents data that are persistent (long-lived) and transactional. In the EJB 1.0 specification, support for entity beans is optional. To test the support for entity beans (if implemented), the following properties should be evaluated:

- The implementation of the EntityContext interface.
- Handling of the concurrent access from multiple clients.
- Support for the findByPrimaryKey() methods, used for both the container and the bean managed persistence.
- The entity bean lifecycle.
- The support for reentrancy.

There is no relationship between Pospíšil and the instantly claimed invention, which is based on a new data construct, a "functional enterprise bean."

Hightower (USP 6,510,550 B1)

Hightower is directed toward "an application development tool [] configured for providing intermittent connectivity support in an application that uses distributed objects. The application development tool includes an intermittent connectivity framework having program code for queuing method calls to a distributed object, as well as an intermittent connectivity component generator having program code for converting an existing

interface for a distributed object into an intermittent connectivity component configured to forward method calls to the intermittent connectivity framework for queuing." See Summary of the Invention.

Hightower describes the need for that invention.

"[I]n order to enable an application running on a mobile computer to participate in a detached session using current technology, it is generally necessary to develop non-trivial special code to write session information to the mobile computer's disk and then synchronize with the enterprise application once the mobile computer is reconnected to the home office." See col. 3, ll. 14-20.

The Office Action argues that motivation could be found in Hightower at col. 8, ll.42-45 (emphasized below).

Under normal (i.e., connected) operating conditions, such as when client computer 50 is connected to a LAN docking station that enables communication with enterprise application 54, a method call for a particular distributed object would generally be directed by local application instance 52 to proxy stub 42, which would in turn forward the method call to enterprise application 54 through transport 56. Enterprise application 54 would then attempt to retrieve the requested distributed object from, for example, a distributed object server (not shown), and then would return the retrieved distributed object to proxy stub 42 through transport 56. In this embodiment, IC bean wrapper 40 provides intermittent connectivity support for those occasions when proxy stub 42 is unable to communicate with enterprise application 54, such as when client computer 50 is not connected to the system on which enterprise application 54 is running.

Still referring to FIG. 5, IC bean wrapper 40 includes instructions for monitoring the activities of proxy stub 42. Upon detecting from proxy stub 42 that method calls for distributed objects cannot be made (e.g., client computer 50 is disconnected), IC bean wrapper 40 forwards the unsatisfied method calls to IC call manager framework 14. IC call manager framework 14 then queues the unsatisfied calls until connectivity with enterprise application 54 is restored. Calls may be queued, for example, in a data store (not shown) resident in a memory of client computer 50 to which IC bean wrapper 40 has read/write access. Once IC bean wrapper 40 detects a connection between client computer 50 and enterprise

application 54, IC call manager 14 processes the queued calls through transport 56 to synchronize local application instance 52 with enterprise application 54 (using, for example, techniques described above). The present embodiment thus permits the user of client computer 50 to continue using local application instance 52 even in the absence of a connection to enterprise application 54.

As can be seen from the excerpted portion, Hightower does not disclose or suggest a functional enterprise bean.

Assuming, *arguendo*, Pospíšil describes the functional bean, it is respectfully submitted that Pospíšil would not, at the time the present invention was made, have suggested or motivated one of ordinary skill in the art that it could be combined with Hightower to achieve the instantly rejected claims. To motivate or suggest that a reference could be combined with another, there must be a showing within the references themselves, that one could be combined with the other. The law clearly states that a hindsight-based reconstruction of the claims is impermissible because almost all inventions are combinations of previously existing things. Applicants respectfully request reconsideration.

Independent claims 7 and 10, are also rejected based on the same combination of references and therefore, Applicants incorporate by reference the above discussion in response to those rejections.

Rejection of claims 18 and 19 under 35 U.S.C. §103(a)
over Nally et al (USP 6,298,478 B1) in view of Hightower

The Office Action rejected claims 18 and 19 over Nally in view of Hightower (described above). As to independent claim 18, the Office Action characterized that EJB 500 in Nally was a "functional bean". Further, the Office Action characterized that "EJB Object 510a and versions of its entity bean, where three versions 520a, 521a, and 522a" are the same as a pool of scarce system resources. Applicants respectfully disagree with this characterization. In the instant application, Applicants describe servicing scarce resources as follows.

"A functional bean provides a functional unit that utilizes one or more scarce resources—such as socket connections to a back-end system, or a memory intensive

resource—for its functionality and hence needs to be shared across the clients that need such resources. The number of functional bean instances can be determined by the number of scarce resources available to the application and can be configured by an application deployer. Load balancing multiple client requests across available functional bean instances can be performed by the container or Service Manager." See p. 14, ll. 22-28.


In Nally, the figures 6 and 6A as well as the accompanying text states that the versions are transactions, but they do not model resources such as socket connections to a back-end system, memory intensive resources etc. Further, there is no suggestion that Nally could be combined with Hightower. Accordingly, Applicants respectfully submit that Nally did not anticipate or render obvious (in combination with Hightower) claim 18.

Because all the independent claims are believed to be patentable, other rejected claims, which are dependent on these base claims, are also patentable. Reconsideration is respectfully requested.

Conclusion

Applicants have addressed all issues raised and all grounds of rejection present in the Office Action. This response does not require any additional fee. Applicants respectfully request reconsideration and an early notice of allowance.

Respectfully Submitted,


(44,602)
Naren Chaganti, Esquire (Reg. No.)
713 The Hamptons Lane
Town & Country, MO 63017
naren@chaganti.com
(650) 248-7011 phone
Attorney for Applicants.

Certificate of Faxing

I certify that on the date below this paper was faxed to the USPTO fax number (703) 746-7239.

Date: July 23, 2003


Naren Chaganti.